

Nintendo GameCube™ PAL Game Development Guidelines

Version 0.2

Table of Contents

1. Introduction.....	4
2. Supporting Multiple Languages.....	4
2.1 Language Settings.....	4
2.1.1 API.....	4
2.1.2 Specifying the Language Setting from the Application.....	5
2.1.3 Game Disc Banner and Memory Card Banners.....	6
2.2 European Instruction Booklets and Software Packaging	7
3. Setup.....	7
3.1 Preparing the PAL Development Hardware	7
3.2 Preparing the Television.....	7
3.3 Using Special Cables	8
3.4 Development Hardware Settings.....	10
3.4.1 Configuring the AMC DDH.....	10
3.4.2 Configuring the NPDP-GDEV	12
3.4.3 Configuring the NPDP-GBOX	13
3.4.4 Configuring the NPDP Console	13
3.4.5 NR Reader.....	14
4. The Video Interface.....	14
4.1 About PAL	14
4.1.1 Differences between PAL and NTSC.....	14
4.1.2 PAL System EFB and XFB Sizes	15
4.1.3 PAL Output	16
4.2 About the EU RGB60 Mode	17
4.2.1 Switching between PAL Mode and EU RGB60 Mode.....	17
4.2.2 Requirements for EU RGB60 Mode.....	18
4.2.3 Recommendations for EU RGB60 Mode	19
4.2.4 Flowchart for Switching between PAL and EU RGB60 Mode	20
4.3 Other Reference Material	21
5. Cautions	22
5.1 Setting the IPL Mode	22
5.2 Known Hardware Bugs in RGB Encoder.....	22
5.3 Resolving the Lockup Issue with the OSResetSystem Function.....	22
5.4 Known Issues with the GXSetDispCopyYScale Function.....	22
5.5 Known Issues with the Nintendo GameCube Executable File <code>videomode.elf</code>	23
5.6 Using a PAL NPDP Console with a USA / JPN Switch.....	24
5.7 Selecting an SDK	24

List of Figures

Figure 1: Recommended flow for language setting process	6
Figure 2: Difference between PAL Cable and NTSC Cable.....	8
Figure 3: NPDP-GDEV and NPDP-GBOX DIP Switch Combinations	12
Figure 4: Flowchart for Switching between PAL Mode and EU RGB60 Mode	20

List of Tables

Table 1: Summary of Language Settings Menu Usage Guidelines	5
Table 2: AV Cable Usage	8
Table 3: SCART-RGB Cable	9
Table 4: Description of NPDP-GDEV and NPDP-GBOX DIP Switch Settings.....	12
Table 5: Differences between PAL and NTSC.....	14
Table 6: Difference between EFB Size and XFB Size for PAL and NTSC.....	15
Table 7: Message Samples for PAL/EU RGB60 Mode Switching	21

Revision History

Version	Revision Date	Description of Revisions
0.2	2/20/02	<ul style="list-style-type: none"> - Modified paragraph 2.1.2 for clarification - Modified Table 1 and removed last row - Added Figure 1 and its description - Modified DDH procedure, paragraph 3.4.1, to include DAC switching - Slightly modified user query message recommendations in paragraph 4.2.1 - Divided paragraph 4.2.2 into sub-paragraphs for clarification - Modified the middle block in Figure 4 (paragraph 4.2.4)
0.17	2/4/02	<ul style="list-style-type: none"> - Revised paragraph 2.2 - Added a paragraph to 4.2 "About the EU RGB60 Mode" - Added image checks to 4.2.2 "Requirements for EU RGB60 Mode" - Added example to 5.2 "Known Hardware Bugs in RGB Encoder" - Added paragraph 5.7 "Selecting an SDK"
0.15	12/24/01	<ul style="list-style-type: none"> - Revised paragraphs 2.1.3, 3.2, 3.3, 3.4.1, 3.4.2, 3.4.3, 3.4.4, 3.4.5, 4.2, 4.2.1, 4.2.2, 4.2.4, 4.3, 5.1, 5.2, 5.3, 5.4, and 5.5 - Added paragraph 5.6
0.14	12/05/01	<ul style="list-style-type: none"> - First NOA Release

1. Introduction

This document describes how to develop PAL applications for the Nintendo GameCube™. This document is intended for developers who have experience in developing NTSC applications. If you are developing Nintendo GameCube software for the first time, you may need to refer to additional documentation.

2. Supporting Multiple Languages

2.1 Language Settings

Many different languages are spoken in regions where the PAL television format is used. When users in such a region purchase a Nintendo GameCube (PAL), they must first select a language. The Nintendo GameCube then saves this information to its SRAM. Users can select English, German, French, Spanish, Italian, or Dutch. These languages are called the “six standard languages.”

The language setting data is used to determine the language in which menu items are displayed on the Nintendo GameCube IPL Menu Screen. Once selected, users can change the saved language setting using the Nintendo GameCube IPL Menu Screen. Applications can also detect the language setting information through the API, and use it in the application. See section 2.1.1 “[API](#)”, for information about the API.

2.1.1 API

Using the `OSGetLanguage` function, applications can detect the language setting selected by the user from the Nintendo GameCube. You can also change the language setting by using the `OSSetLanguage` function.

```
u8 OSGetLanguage( void );           //Get language setting information
```

The return values for `OSGetLanguage` are as follows:

```
#define OS_LANG_ENGLISH    0u      // English
#define OS_LANG_GERMAN     1u      // German
#define OS_LANG_FRENCH     2u      // French
#define OS_LANG_SPANISH    3u      // Spanish
#define OS_LANG_ITALIAN    4u      // Italian
#define OS_LANG_DUTCH      5u      // Dutch
```

Do not change language setting information (by modifying the content of SRAM) in production versions of applications. However, during development you may need to change the language setting for testing purposes. In these situations, use the `OSSetLanguage` function.

```
void OSSetLanguage( u8 language );   // Change language setting information
```

In the argument, specify one of the return values listed in the description of the `OSGetLanguage` function.

2.1.2 Specifying the Language Setting from the Application

PAL applications should detect the Nintendo GameCube language settings information, stored in the Nintendo GameCube, at startup. If the application supports the language stored in the Nintendo GameCube language setting, all text messages in the application should be displayed in that language. If the language is not supported by the application, the application should display supported languages and ask player to select one of the languages.

However, do not change language setting information saved in the Nintendo Game Cube from the application. Application-specific language settings should be saved to the Nintendo GameCube Memory Card as Application Language Setting Data.

To determine whether to include a language settings menu in your application, consider the following items:

- **The application supports only one of the six standard languages.**
The language setting cannot be changed, so no language settings menu is needed in the application.
- **The application supports only some of the six standard languages (two to five languages).**
If the application supports only some of the standard languages (two to five languages), we recommend that you include a language settings menu in the application.
For example, a player who normally uses Dutch might want to select French (or German) when playing a game that does not support Dutch.
- **The application supports all six standard languages.**
If the application supports all six standard languages, we recommend that you include a language settings menu in the application.
A player may want to select a language other than the one they normally use, depending on the application.
- **The application supports multiple languages, including languages other than the six standard languages.**
If the application supports multiple languages, including languages other than the six standard languages, a language settings menu is required.
Languages that cannot be selected with the Nintendo GameCube need to be set in the application.

The preceding descriptions are summarized in the following table:

Supported Languages	One of the six standard languages	Two to five of the six standard languages	All six standard languages	Language(s) other than the six standard languages
Language Settings Menu Requirements	n/a	Recommended	Recommended	Required

Table 1: Summary of Language Settings Menu Usage Guidelines

Refer to the process flow below showing the setting of languages in the application.

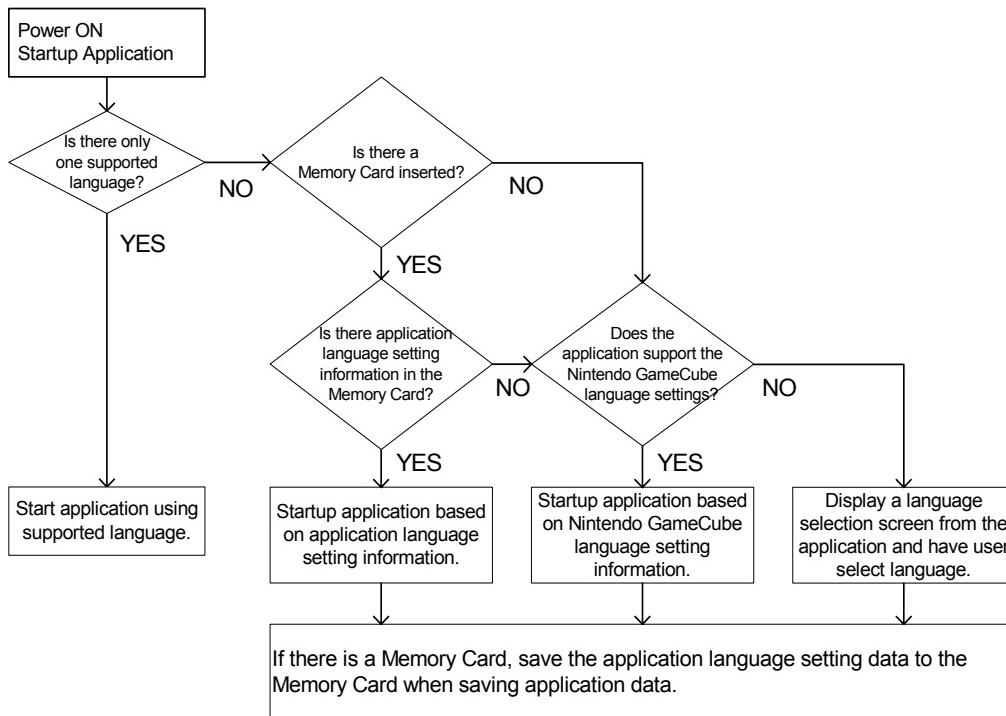


Figure 1: Recommended flow for language setting process

Use the language setting information, not only with the application messages, but also use this information with the Game Disc and Memory Card error messages, and the messages for switching between PAL and EU RGB60.

Even if the application does not support multiple languages, it is recommended that you provide multiple language support for the Game Disc and Memory Card error messages, to the extent that is possible.

In addition, if the application has its own Application Language Setting Information, it is recommended that you give this priority over the Nintendo GameCube language setting information, when displaying the error messages.

For more information about the translation of messages, refer to the included document, "Nintendo GameCube Standard Language Conversion Table" (StandardLanguageConversion*.us.pdf), and section 4 "[The Video Interface](#)".

2.1.3 Game Disc Banner and Memory Card Banners

The main menu of the Nintendo GameCube (PAL) supports the expanded format of the Game Disc banner. In the expanded format of the Game Disc banner you can show every part (`shortTitle`, `shortMaker`, `longTitle`, `longMaker`, `comment`), in all six languages. The expanded format of the Game Disc banner is supported by the December 12, 2001 release of the SDK (`makebanner2.exe`).

The format of the comment portion of the Memory Card banner differs from that of the Game Disc banner, and does not support multiple languages. For the application, change the comment saved to the Memory Card, based on the language setting information. As long as there are no problems, match the languages supported by the Game Disc banner and Memory Card banner with those supported by that particular application.

2.2 European Instruction Booklets and Software Packaging

Refer to "Nintendo GameCube Software Packaging Guidelines" for information on Instruction Booklets and software packaging. For the official language display of each part, refer to the included document, "Nintendo GameCube Standard Language Conversion Table" (StandardLanguageConversion*.us.pdf).

3. Setup

3.1 Preparing the PAL Development Hardware

Nintendo GameCube development hardware supports PAL development mode by attaching the PAL video encoder to the video board of the DDH, NPDP-GDEV, or NPDP-GBOX. Video boards made to support PAL development mode will also support NTSC and MPAL.

To upgrade your NTSC development system to a system that supports NTSC, MPAL, and PAL, contact NOA Technical Services.

NPDP Consoles and NR Readers for PAL are also available. Please contact NOA for more information.

3.2 Preparing the Television

Use a PAL television when developing PAL applications. You need a television that uses an RCA jack for its video input terminal (the normal terminal for connecting an AV cable) or one with a SCART terminal. To verify RGB output from the Nintendo GameCube, use a television with a SCART terminal that allows RGB input (SCART-RGB terminal).

If creating an application supporting EU RGB60 mode as described in section 4.2 "[About the EU RGB60 Mode](#)", use a television that handles SCART-RGB input, and also follows PAL60 standards.

3.3 Using Special Cables

This section describes how to handle cables used primarily in regions that use the PAL television format. If you work in both PAL and NTSC development environments, you should be aware that selecting the appropriate cable may be confusing.

- **AV Cable**

To display in PAL mode, you need to use different AV cables depending on which hardware you use (see Table 2). You can distinguish PAL and NTSC AV cables by looking at the color of the line on the cable. The PAL cable has a blue line printed on it, and the NTSC cable has a white line printed on it (See Figure 1 below).

Hardware that Uses the NTSC AV Cable
DDH, NPDP-GDEV, NPDP-GBOX (Hardware supporting PAL and NTSC)
Hardware that Uses the PAL AV Cable
PAL NR READER, PAL NPDP CONSOLE

Table 2: AV Cable Usage

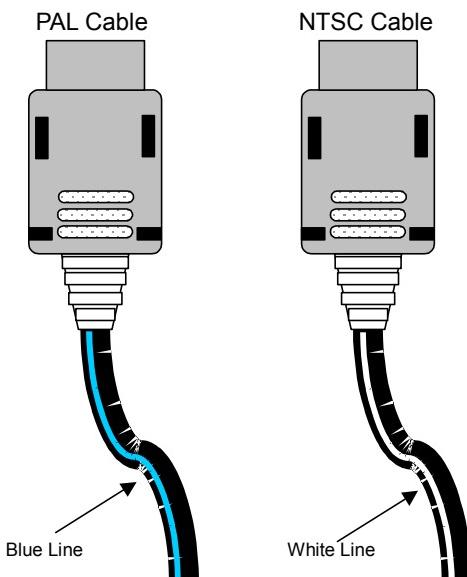


Figure 2: Difference between PAL Cable and NTSC Cable

- **S Video Cable**

You cannot use an S Video cable with the PAL version of Nintendo GameCube.

- **RGB Cable**

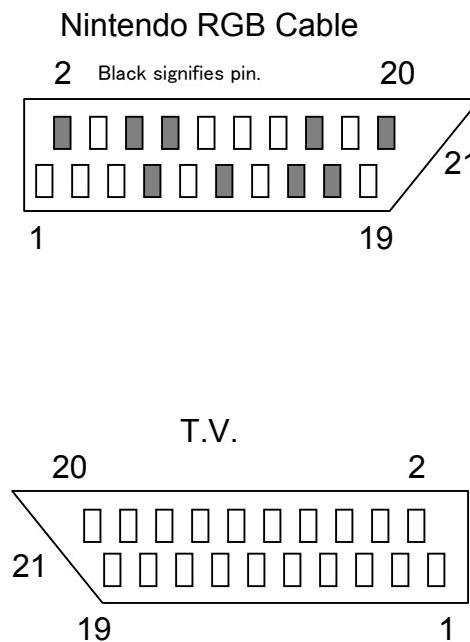
To display in RGB output mode from Nintendo GameCube, connect the Nintendo GameCube to a television with SCART-RGB input using an RGB cable. Some televisions may have a SCART terminal that only supports SCART-AV input (composite video input). With such televisions, you will only be able to obtain image quality equivalent to composite video output and will not be able to see true RGB-quality images.

The RGB Cable can be used with the PAL Nintendo GameCube, PAL NPDP Console, and PAL NR Reader.

You can only display in EU RGB60 mode when using an RGB cable. For more details on EU RGB60 mode, see section 4.2 “[About the EU RGB60 Mode](#)”.

Pin #	Description
1	Audio Output (right)
2	
3	Audio Output (left)
4	Audio GND
5	Blue GND (RGB)
9	Green GND (RGB)
10	Clock Output
12	Data Output
13	Red GND (RGB)
14	Data GND
18	Blank Signal GND
19	Composite Video Output
20	Comp. Video/Brightness Input
21	GND

Table 3: SCART-RGB Cable



3.4 Development Hardware Settings

This section describes the settings for the various development hardware.

3.4.1 Configuring the AMC DDH

Connect the TV to the DDH with an NTSC AV cable. If you are using DDK3.0 or later, use the following procedure to switch to PAL development mode:

1. Set the DDH video mode to PAL:

```
% loadrun.bat videomodeD.elf -a 1
```

2. Set the Game Disc Country Code to "eu":

```
% setcountrycode eu
```

3. Set the DDH Japan/USA switch to USA, and switch DAC to PAL:

(The "setcountry" and "setdac" commands are supported starting from AMCDK ver. 3.0.2 patch #2. If these commands are used, the DDH does not have to be rebooted, and the following steps do not need to be executed. See the DDK manual for more details.)

- a) Execute stop.

- b) Edit the country code field in D:\amc.cfg (the D drive is the emulation drive), and set it to usa:

```
[118] 00000000 // default country code 0 usa, 1 japan
```

- c) Reboot the DDH.

4. Set the DDH to PAL mode:

- a) Reboot the DDH (if necessary).

- b) Press the Enter key once from the DDH Config console to bring up the amc> prompt.

- c) Execute the next command:

```
amc> mem set /w 4010024
```

By executing this command, a hyphen is displayed in the current memory address value and the command prompt awaits input:

```
04010024: 0009-
```

If you input 1 at this point, the command prompt will wait for input of the next address. (The same procedure applies to values other than 0009.)

```
04010024: 0009-1 ← Input 1
```

```
04010026: 0002-
```

Finally, input a period to exit the DDH Config console.

```
04010026: 0002-. ← Input period
```

By following the procedure outlined above, you can set the DDH SRAM data and Game Disc Country Code setting to PAL development mode.

NOTE: The DDH will remain in PAL mode only as long as power is maintained. If you reboot the DDH, you will need to repeat Step 4, above.

To switch a DDH that has been configured for PAL development back to NTSC development mode, use the following procedure:

1. Set the DDH Japan/USA switch to USA or Japan, and switch DAC to NTSC.
(The "setcountry" and "setdac" commands are supported starting from AMCDKK ver. 3.0.2 patch #2. If these commands are used, the DDH does not have to be rebooted, and the following steps do not need to be executed. See the DDK manual for more details.)
 - a) Execute stop.
 - b) Edit the country code field in D:\amc.cfg (the D drive is the emulation drive), and set it to usa mode or japan mode:
[118] 00000000 // 0 is USA
[118] 00000001 // 1 is Japan
 - c) Reboot the DDH.
2. Set the DDH video mode to NTSC.
% loadrun.bat videomodeD.elf -a 0
3. Set the Game Disc Country Code to "us" or "jp".
% setcountrycode us
Or
% setcountrycode jp

By following the procedure outlined above, you can set the DDH SRAM data and the Game Disc Country Code setting to NTSC development mode.

3.4.2 Configuring the NPDP-GDEV

Connect the TV to the NPDP-GDEV with an NTSC AV cable. Switch to PAL development mode using the following procedure:

Warning: Turn OFF power to the NPDP-GDEV before operating DIP switches.

1. Set DIP switch.

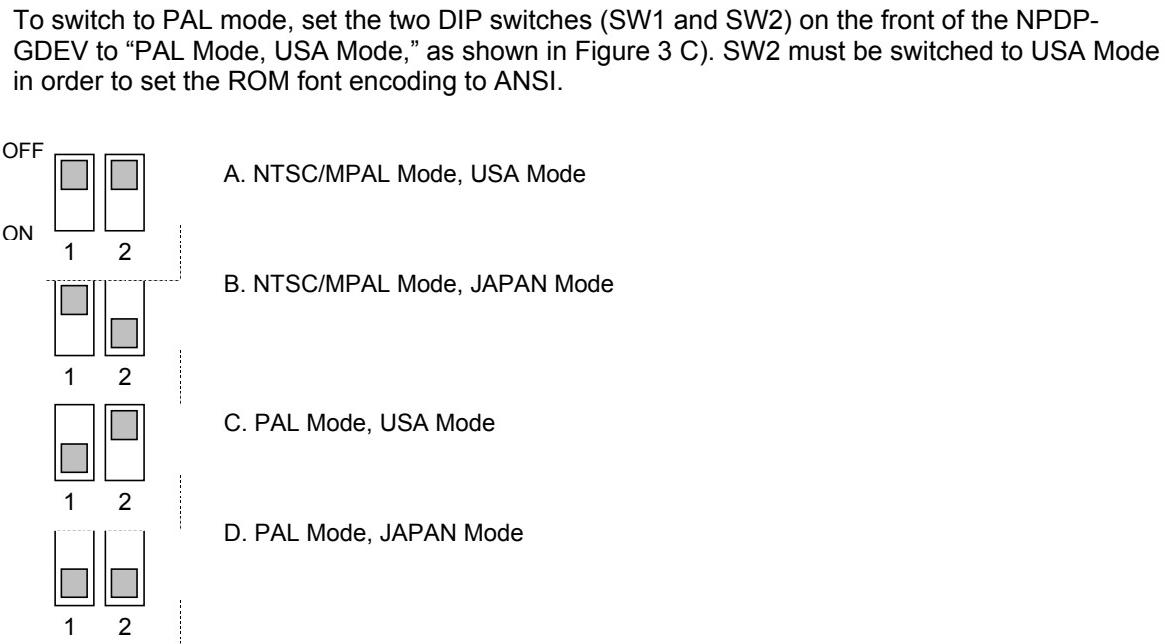


Figure 3: NPDP-GDEV and NPDP-GBOX DIP Switch Combinations

Switch	Status	Description
SW1	ON	PAL Mode
	OFF	NTSC/MPAL Mode
SW2	ON	JAPAN Mode
	OFF	USA Mode

Table 4: Description of NPDP-GDEV and NPDP-GBOX DIP Switch Settings

2. Turn ON power to the NPDP-GDEV.
3. Set the NPDP-GDEV video mode to PAL:

```
% odrun.bat videomodeD.elf -a 1
```
4. Set the Game Disc Country Code to "eu":

```
% setcountrycode eu
```

By following the procedure outlined above, you can set the NPDP-GDEV SRAM data and Game Disc Country Code setting to PAL development mode. Even if you set `videomode` to NTSC, the PAL IPL Menu Screen will display (although the animation will be faster than normal).

To switch an NPDP-GDEV that has been configured for PAL development back to NTSC development mode, use the following procedure:

1. Turn OFF power to the NPDP-GDEV and set the DIP switches to A or B, as shown in Figure 2. (A is USA mode and B is JAPAN mode).
2. Set the NPDP-GDEV video mode to NTSC:
 % odrun.bat videomodeD.elf -a 0
3. Set the Game Disc Country Code to "us" or "jp".
 % setcountrycode us
 OR
 % setcountrycode jp

By following the procedure outlined above, you can set the NPDP-GDEV SRAM data and Game Disc Country Code setting to NTSC.

3.4.3 Configuring the NPDP-GBOX

Connect the TV to the NPDP-GBOX with an NTSC AV cable. Use the DIP switches to switch to PAL development mode.

Warning: Turn OFF power to the NPDP-GBOX before operating DIP switches.

Switch the two DIP switches on the front of the NPDP-GBOX to "PAL Mode, USA Mode," as shown in Figure 2 C. SW2 must be switched to USA Mode in order to set the ROM font encoding to ANSI.

To switch an NPDP-GBOX from PAL mode back to NTSC mode, configure the dip switches as shown in Figure 2 "A" or "B".

3.4.4 Configuring the NPDP Console

Connect the TV to the NPDP Console with a PAL AV cable or RGB cable. Switch to USA mode using the switch on the left side of the NPDP Console.

If you use a PAL NPDP Console that has a USA / JPN switch on the left side, you will need to observe the precautions described in paragraph 5.6 "[Using a PAL NPDP Console with a USA / JPN Switch](#)".

The procedure for changing the Country Code setting varies depending on the file type.

To set the Country Code for a GCM file:

- Set the Country Code to "Europe" using the Nintendo GameCube Master Editor software.

To set the Country Code for a DLF or DPF file:

1. Use setcountrycode to change the Country Code setting.
2. Execute the application. (If you use setcountrycode after starting the application, the language settings may not necessarily be changed. Therefore, change the Country Code setting before executing the application.)

For security reasons, applications on an NPDP cartridge will not run on an NPDP Console if the Country Code setting in the application is not set to eu.

3.4.5 NR Reader

Connect the TV to the PAL NR Reader using a PAL AV cable or RGB cable.

In the gcm file you write to the NR Disc, set the Country Code to Europe (eu). Use the Nintendo GameCube Master Editor software to change the country code. For security reasons, the application will not run if the Country Code in the gcm file on the NR Disc does not match the Country Code in the NR Reader.

4. The Video Interface

This section describes the video format for Nintendo GameCube (PAL).

4.1 About PAL

4.1.1 Differences between PAL and NTSC

There are two major differences between the PAL system and the NTSC system.

1. Timing

With the PAL system, 50 fields can be expressed in one second. In other words, one field corresponds to 20 ms. Compared to the NTSC system's 16.6 ms/field, the PAL system requires 19.88% more time to display one field.

2. Number of Lines

The PAL system uses 625 lines for one field. Among these 625 lines, 574 are active. "Active" means that a picture can be drawn in these lines (the remaining 48 lines are used for vertical retrace). The NTSC system, however, uses 525 lines for one field. Among these 525 lines, 480 are active.

	Fields/Second	Time for One Field	Active Lines
PAL	50	20.0 ms	574
NTSC	60	16.6 ms	480

Table 5: Differences between PAL and NTSC

4.1.2 PAL System EFB and XFB Sizes

The PAL system has more scan lines than the NTSC system. Thus, generally you need to make the XFB vertical size for the PAL system larger than the XFB vertical size for the NTSC system.

When the EFB and XFB sizes are both 480 on an NTSC system, the vertical display for the average NTSC television will be over scanned (the application screen will not fit properly within the television screen).

When the EFB and XFB sizes are both 528 on a PAL system, the vertical display will be about the size of the screen for the average PAL television.

The following table outlines differences between generally used EFB and XFB sizes on NTSC and PAL systems:

	EFB Size	XFB Size
NTSC	640x480	640x480
PAL	640x528 (Maximum EFB Size)	640x528

Table 6: Difference between EFB Size and XFB Size for PAL and NTSC

When copying a rendered image from EFB to XFB with Nintendo GameCube, you can use the GX feature and scale in the vertical display. For more details, refer to Chapter 5 of the document *Video Interface Library (VI)* in the Nintendo GameCube SDK, Version 1.0 or later.

4.1.3 PAL Output

In order to output to a PAL system, you need to change the `GXRendModeObj` structure member `viDisplayMode` to PAL.

Specify `VI_TVMODE_PAL_INT` for PAL interlaced mode, and `VI_TVMODE_PAL_DS` for PAL non-interlaced mode.

The following sample code displays a 640x528 PAL interlaced screen to the television, using 640x528 EFB.

```
GXRendModeObj GXPal528Int =
{
    VI_TVMODE_PAL_INT,                      // viDisplayMode
    640,                                     // fbWidth
    528,                                     // efbHeight
    528,                                     // xfbHeight
    (VI_MAX_WIDTH_PAL - 640)/2,              // viXOrigin
    (VI_MAX_HEIGHT_PAL - 528)/2,             // viYOrigin
    640,                                     // viWidth
    528,                                     // viHeight
    VI_XFBMODE_DF,                          // xFBmode
    GX_FALSE,                                // field_rendering
    GX_FALSE,                                // aa

    // sample points arranged in increasing Y order
    6, 6, 6, 6, 6, // pix 0, 3 sample points, 1/12 units, 4 bits each
    6, 6, 6, 6, 6, // pix 1
    6, 6, 6, 6, 6, // pix 2
    6, 6, 6, 6, 6, // pix 3

    // vertical filter[7], 1/64 units, 6 bits each
    0,      // line n-1
    0,      // line n-1
    21,     // line n
    22,     // line n
    21,     // line n
    0,      // line n+1
    0       // line n+1
};
```

4.2 About the EU RGB60 Mode

In PAL mode the field frequency is 50 Hz and there are 574 active scan lines. In EU RGB60 mode the field frequency is 60 Hz and the 480 active scan lines are output as an RGB signal.

If you are going to be using EU RGB60 mode, use patch 4 for the 09/08/2001 version of the Nintendo GameCube SDK along with the EU RGB60 patch, or a version of the Nintendo GameCube SDK dated 12/12/2001 or later.

By connecting an RGB cable, a TV that has a 21-pin SCART-RGB input terminal and supports the PAL60 standard can display images in both PAL mode and EU RGB60 mode.

Language support for EU RGB60 mode does not necessarily have to be the same as that for PAL mode. For example, you could have a combination in which PAL mode supports three languages and EU RGB60 mode only supports English. (In other words, you would be able to use an application created for the US market as is, in EU RGB60 mode.)

Note: All PAL mode applications do not have to support EU RGB60 mode.

If the application is for a region where the PAL television format is used, do not provide support for progressive scan mode. Completely remove any functionality used to change the display to progressive scan mode. (See “Nintendo GameCube Video Guidelines” for more information on progressive scan mode.)

4.2.1 Switching between PAL Mode and EU RGB60 Mode

The operations for switching between PAL mode and EU RGB60 mode on Nintendo GameCube are very similar to those for switching between interlaced mode and progressive scan mode. Before performing the operations for switching between PAL mode and EU RGB60 mode, you must carry out a two-step check (see section 4.2.4 “[Flowchart for Switching between PAL and EU RGB60 Mode](#)”).

The EU RGB60 verification flag uses one bit of the Nintendo GameCube SRAM. Nintendo GameCube uses this flag as one of the conditions when determining whether to display a message to users asking whether or not they want to switch from PAL mode to EU RGB60 mode. Use the `OSGetEuRgb60Mode` function to check the EU RGB60 verification flag, and use the `OSSetEuRgb60Mode` function to set the EU RGB60 verification flag to ON or OFF. The default factory setting for the EU RGB60 verification flag is OFF.

In the actual process, the application should always be displayed in PAL mode when starting up. Then perform a check to see if the B Button is being pressed, or if the EU RGB60 verification flag is set to ON. Check whether a user is pressing the B Button from the time the Nintendo GameCube logo is displayed until the first application screen is displayed (in other words, the very first image that your application displays).

Finally, display a user query message in PAL mode if one of the following requirements is met:

- The user is pressing the B Button when game processes begin.
- The EU RGB60 verification flag is set to ON.

For example, you could display a message such as “Would you like to display in 60 Hz mode?” (For more information, see section 4.2.2. “[Requirements for EU RGB60 Mode](#).“)

If the EU RGB60 verification flag is set to ON (1), the user message should be displayed automatically. If set to OFF (0), the user message does not display unless the user presses B Button when the application starts.

We recommend that you display the user query message in the language specified in the Nintendo GameCube language setting information. To obtain the language setting specified in the Nintendo GameCube, use the `OSGetLanguage` function. (See section 2.1 “[Language Settings](#).” Also refer to [Table 7](#) for sample user messages.)

The default position of the cursor for this message is “YES”. If “YES” is selected, the EU RGB60 verification flag is set to ON and Nintendo GameCube will display the game in EU RGB60 mode from that point on. If the user selects “NO,” the EU RGB60 verification flag is set to OFF and Nintendo GameCube will display the application in PAL mode from that point forward.

If the time required to load data at application startup is long, the user query message should be displayed while waiting for the game’s title screen to be displayed. Set a timer for the message and display it for around ten seconds. If the user does not make a selection and a timeout occurs, carry out the operations for the selection where the cursor was left. We recommend that the application perform its various data load processes in the background during this time.

If the time required to load data at application startup is short, the user query message should be displayed together with the game title. Do not automatically switch to EU RGB60 mode until the user makes a selection.

The query message should be displayed using an appropriate method (during data load, with title screen, and so on).

4.2.2 Requirements for EU RGB60 Mode

4.2.2.1 Naming the Mode

In all consumer-oriented text, use the term “60 Hz mode,” not “EU RGB60 mode.” Similarly, use the term “50 Hz mode” instead of “PAL mode.” (Use “60 Hz mode” and “50 Hz mode” in all on-screen messages and instruction booklet text, instead of “EU RGB60 mode” and “PAL mode.”)

4.2.2.2 Describing the Mode in the Instruction Booklet

In the Instruction Booklet for applications supporting EU RGB60 mode, include a description of the following three items:

1. Description of the two modes

Example: 50 Hz mode has been the standard PAL display mode for past game consoles. By using an RGB Cable to connect Nintendo GameCube with a television that supports PAL60 and has a SCART-RGB input terminal, you can display the game in 60 Hz mode, which provides a smoother and more vivid display than was possible in the past.

2. How to switch between the two modes

Example: Press and hold the B Button after the Nintendo GameCube logo is displayed (or while it is displayed) until the message “Would you like to display in 60 Hz mode?” appears on the screen.

3. Cautions

Example: If you want to play the game in 60 Hz mode, use an RGB Cable to connect Nintendo GameCube with a television that supports PAL60 and has a SCART-RGB input terminal. Correctly configure the television’s display switch and other settings for RGB display.

In order to make the selection of EU RGB60 mode easy for the user, make the default position for the cursor “YES” for the message that asks, “Would you like to display in 60 Hz mode?”

4.2.2.3 Verifying the Display

For applications that support EU RGB60 mode, check that the image is displayed correctly in the following three scenarios:

- Connect Nintendo GameCube using an AV cable and output a 50 Hz composite-video signal.
- Connect Nintendo GameCube using an RGB cable and output a 50 Hz RGB signal.
- Connect Nintendo GameCube using an RGB cable and output a 60 Hz RGB signal.

For details on working with the various cables, see Section 3.3 “[Using Special Cables.](#)”

The process for switching the display to EU RGB60 mode is very similar to the process for switching to progressive scan mode. Applications for regions where the PAL television format is used do not support progressive scan mode, so completely remove any processes used for switching to this mode.

4.2.3 Recommendations for EU RGB60 Mode

Carry out the switch between PAL mode and EU RGB60 mode when the application starts, and save the settings information to the Nintendo GameCube SRAM as the EU RGB60 verification flag. Only perform this switch process between PAL mode and EU RGB60 mode at this time.

4.2.4 Flowchart for Switching between PAL and EU RGB60 Mode

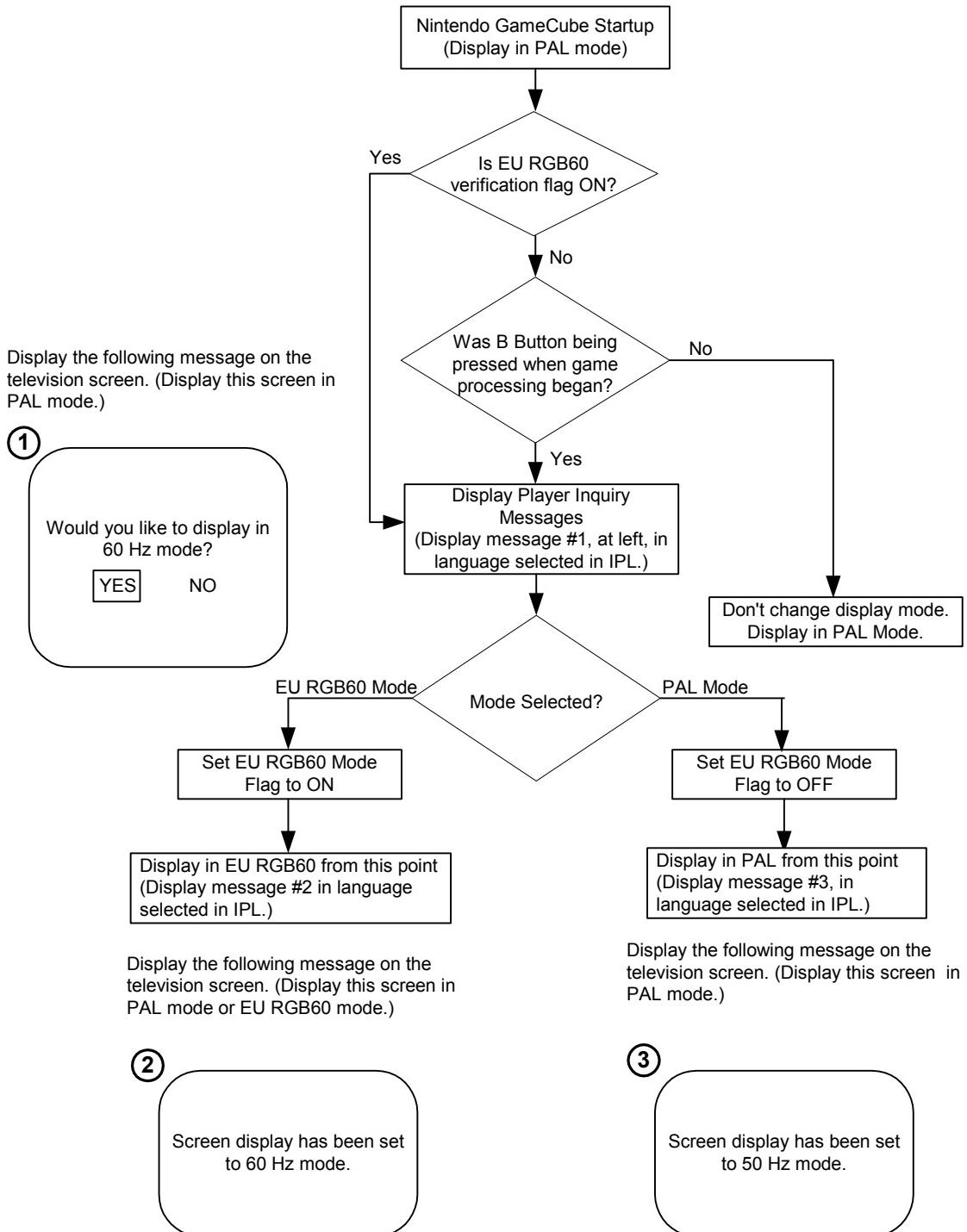


Figure 4: Flowchart for Switching between PAL Mode and EU RGB60 Mode

English	French	German
Would you like to display in 60 Hz mode?	Voulez-vous afficher en mode 60 Hz?	Soll die Darstellung in den 60Hz-Modus umgeschaltet werden?
Yes	Oui	Ja
No	Non	Nein
Screen display has been set to 60 Hz mode.	L'affichage est maintenant en mode 60 Hz.	Die Bildschirmsdarstellung erfolgt im 60Hz-Modus.
Screen display has been set to 50 Hz mode.	L'affichage est maintenant en mode 50 Hz.	Die Bildschirmsdarstellung erfolgt im 50Hz-Modus.
Italian	Spanish	Dutch
Visualizzare nella modalità 60 Hz?	¿Deseas ver el juego en modo 60 Hz?	Wil je weergave in 60 Hz mode?
Si	Sí	Ja
No	No	Nee
Lo schermo è stato impostato nella modalità 60 Hz.	La pantalla se ha configurado en modo 60 Hz.	De beeldweergave is ingesteld op 60 Hz mode.
Lo schermo è stato impostato nella modalità 50 Hz.	La pantalla se ha configurado en modo 50 Hz.	De beeldweergave is ingesteld op 50 Hz mode.

Table 7: Message Samples for PAL/EU RGB60 Mode Switching

4.3 Other Reference Material

For more details on the video interfaces described here, refer to the *Video Interface Library (VI)* document in the Nintendo GameCube SDK, Version 1.0 or later.

5. Cautions

5.1 Setting the IPL Mode

As mentioned in paragraph 2.1 "[Language Settings](#)", you can set up the six standard languages from the main menu options with the Nintendo GameCube (PAL). In order to display this main menu with the DDH or NPDP-GDEV, the IPL mode needs to be in "Production IPL Mode". The IPL mode can be switched by using the Nintendo GameCube executable file, bootmode.elf:

- Set IPL to production IPL mode:
`%loadrun.bat bootmodeD.elf -a 1`
- Set IPL to development IPL mode:
`%loadrun.bat bootmodeD.elf -a 0`

Warning: If you are using a version of DEVKIT BOOT ROM prior to Version 0.93a, problems occur when you run the Nintendo GameCube executable file videomode.elf. Therefore, avoid using the procedure described above. For additional details, see section 5.5 "[Known Issues with the Nintendo GameCube Executable File videomode.elf](#)."

5.2 Known Hardware Bugs in RGB Encoder

We are currently aware of hardware bugs with the circuit that creates the PAL video RGB data. The two pixels on the right edge of the screen may not display correctly. This problem does not occur if you use a stereo AV cable, but if you are developing a game that supports EU RGB60, you may encounter this bug.

For example, if you run `colorD.elf` (included in the Nintendo GameCube SDK's "videmo") on hardware that has this problem, you can see that the two right-most pixels are white when the yellow screen is rendered

5.3 Resolving the Lockup Issue with the OSResetSystem Function

With version combinations of DEVKIT BOOTROM 0.94a (Beta) and Nintendo GameCube SDK (09/08/2001) + patch3 (or an older patch), the Nintendo GameCube OS locks up after calling:

```
OSResetSystem (OS_RESET_RESTART, ...)
```

This problem has been solved in DEVKIT BOOTROM Version 0.94b (Beta) and Nintendo GameCube SDK (09/08/2001) patch 4.

5.4 Known Issues with the GXSetDispCopyYScale Function

Currently we are aware of three hardware problems related to `GXSetDispCopyYScale`. Issues 2 and 3 have been resolved in Version 12/12/01 of the Nintendo GameCube SDK. However, if you use a version of the SDK that was released prior to the 12/12/01 version, issues 2 and 3 may occur.

1. There are some ratios that are not supported by the hardware when enlarging.

The accuracy of the Y scale in the hardware is low, so a copy from an arbitrary EFB line to an arbitrary XFB line cannot always be assured. For example, if you try to enlarge from 480 to 528, there may not be an appropriate ratio for this and a similar value may be used instead. Values such as 522, 525, 527, 529, 531, or 534 could be used.

This problem is caused by the hardware specifications. You can resolve this issue by changing VI settings with the currently supported line numbers, or using the same VI setting and masking the bottom line in black, where not copied. Additional caution is necessary when working with interlaced mode or frame rendering, since the XFB height cannot be set to an odd number with VI.

In order to maintain the same vertical and horizontal ratio as NTSC on a PAL television, a setting of around 480 to 568 would be appropriate. This translates to 530 with an EFB of 448 lines, and 552 with an EFB of 464 lines. These values are all supported by the hardware, so use these values when determining what ratio to use for your application.

2. There is a calculation error when an argument is used as a floating point.

Even if the calculation is a ratio supported by the hardware, but you specify a floating point for the API argument, a calculation error occurs when the value is passed to the hardware. This causes the result to be off by several lines.

3. `GXSetDispCopyYScale` Bug

Currently we are aware of an error with the return value of `GXSetDispCopyYScale`. It is possible that the correct XFB line number will not be returned.

5.5 Known Issues with the Nintendo GameCube Executable File `videomode.elf`

Be aware of the following points if you are using a development kit (DDH/GDEV) with an Orca board previous to DEVKIT BOOT ROM Version 0.93a, or if you are using an NPDP-GBOX or NPDP Console:

Warning: The development kit will stop working if you use `bootmode.elf` to change the development kit's IPL to "Production IPL Mode," and then run `videomode.elf` and change the video mode to "PAL Mode." When using the production IPL mode, do not change the video mode setting to PAL for any reason. If you carry out this operation and the development kit no longer functions, you will have to return it to Nintendo to be fixed. This problem was fixed in the December 2001 version of the SDK.

You can check the DEVKIT BOOT ROM version with the development kit's console output:

```
--- DEVKIT BOOTROM v0.94b (PRODUCTION MODE: NTSC) ---  
--- DEVKIT BOOTROM v0.93a (PRODUCTION MODE) ---
```

In the preceding display, "v0.xxx" indicates the version number. If the BOOT ROM is a version prior to v0.8 of the GDEV with IPL, only the following is displayed:

```
--- DEVKIT BOOTROM ---
```

5.6 Using a PAL NPDP Console with a USA / JPN Switch

If you use a PAL NPDP Console that has a USA/JPN switch to develop PAL applications, move the switch on the left side of the console to the “USA” position. The Country Code for the file written to the NPDP Cartridge needs to be set to “us”.

This specialized NPDP Console uses the NTSC IPL, and has a PAL video encoder for the video output. Due to the limitations of this configuration, the following points must be taken into consideration.

- **IPL Language Setting Feature**

To check the IPL language setting feature, the PAL IPL is required. However, you cannot use the IPL language setting feature on an NPDP Console that has the NTSC IPL. Until an NPDP Console with a PAL IPL is produced, check the language setting feature with a PAL NR Reader.

- **VI Initialization**

The IPL initializes the VI to NTSC, and the IPL Menu Screen is displayed in NTSC mode. However, when processing is transferred to the application, the application attempts to initialize the VI to PAL. The screen may become corrupted at this time, but this problem will be resolved when an NPDP Console with a PAL IPL is released.

- **Initializing the IPL Settings**

When using a PAL television, if the screen flickers until the application starts (after the IPL starts up), you will not be able to initialize the IPL settings (language, calendar, and so on). If the screen does flicker, connect the NPDP Console to an NTSC television and change any necessary settings.

5.7 Selecting an SDK

The following limitations with the Nintendo GameCube SDK apply when you are developing PAL applications:

- Development of application with EU RGB60 mode:
Patch 4 for the 09/08/2001 version of the SDK, along with the EU RGB60 patch, or a version of the SDK dated 12/12/2001 or later.
- Development of application without EU RGB60 mode:
Patch 4 for the 09/08/2001 version of the SDK or a later version.

Warning: Versions of the SDK prior to those listed above do not support PAL.